1. What do you mean by union compatibility of relations? Explain with example.

The following conditions must be met by the participating relations in order for them to be union compatible.

The two relations must have the same degree, i.e. they must have the same number (set) of attributes.

Dom(A) = Dom(B)

That is to say, the domain (data type) for the relevant characteristics must be the same.

As a result, we may state that any two relations, such as A and B, are union compatible if (and only if) both relations have the same number of attributes and their associated attribute domains are the same (column by column).

Example:

S (Sid, Sname, Marks) and R (Sid, Sname, Marks) are not UNION COMPATIBLE because R has two characteristics and S has three.

Because the domains of name and marks are different, R (Sid, Sname) and S (Sid, Marks) are not UNION COMPATIBLE. Certain texts would be the realm of names, whereas integer and float numbers would be the domain of markings.

R (Sid, Sname) and S (Sid, StudentName) are UNION COMPATIBLE because they both have two attributes and the identical domains; the only difference is the column names.

Step 2:

The operations for which union compatibility of relations is required are listed below.

All tuples from each of the relations are contained in the Union (A B).

It contains all the tuples that are contained in both the relations A and B. Intersection (A B)- It contains all the tuples that are contained in both the relations A and B.

It comprises all the tuples that are present in relation A but not present in relation B.

The set of all concatenated tuples (x, y), where x is a tuple in the relation A and y is a tuple in the relation B, is known as the Cartesian Product (A B). The product of each tuple of relation A and each tuple of relation B is contained in the tuples in the Cartesian product. (It should be noted that the two relations do not have to be union compatible in order to perform the Cartesian product.)

It is critical that the two relations are union compatible in order to conduct these procedures. It's because relations that aren't union compatible, meaning they don't have the same degree of characteristics and share the same domain, might make such actions problematic.

Discuss various set operations of relational algebra with example.

Step 1:

The following are the most common types of set operations:

Union operation

Intersection operation

Set difference or Minus operation

Step 2:

UNION OPERATION: Notation: A S, where A and S are the relations and the sign U denotes the Union operator.

The outcome of the Union operation, denoted by A S, is a relation that generally includes all of the tuples that are contained in A, S, or both, while removing duplicates.

Example-

Take a look at the following two relationships: R and S-

Relation R

|  |  |
| --- | --- |
| ID | Name |
| 10 | Anju |
| 20 | Raju |

Relation S

|  |  |
| --- | --- |
| ID | Name |
| 10 | Anju |
| 30 | Meenu |

Then Relation RUS

|  |  |
| --- | --- |
| ID | Name |
| 10 | Anju |
| 20 | Raju |
| 30 | Meenu |

**Intersection operaton ∩**

Let R and S be two different types of relationships.

The set of all tuples belonging to both R and S is then R S.

Duplicates are automatically eliminated in R S.

Both commutative and associative operations are used in intersection operations.

Example-

Take a look at the following two relationships: R and S-

Relation R

|  |  |
| --- | --- |
| ID | Name |
| 10 | Raju |
| 20 | Ammu |
| 30 | Veena |

Relation S

|  |  |
| --- | --- |
| ID | Name |
| 40 | Ramu |
| 20 | Ammu |

Relation R**∩**S

|  |  |
| --- | --- |
| ID | Name |
| 20 | Ammu |

Difference operator (-)

Let R and S be two different types of relationships.

The set of all tuples belonging to R but not to S is then R – S.

Duplicates are automatically eliminated in R – S.

The operation of difference is associative, but not commutative.

Example-

Take a look at the following two relationships: R and S-

Relation R

|  |  |
| --- | --- |
| ID | Name |
| 10 | Raju |
| 20 | Ammu |
| 30 | Veena |
| 40 | Ramu |

Relation S

|  |  |
| --- | --- |
| ID | Name |
| 20 | Ammu |

Relation R-S

|  |  |
| --- | --- |
| ID | Name |
| 10 | Raju |
| 20 | Veena |
| 30 | Ramu |

What is the purpose of rename operator in relational algebra?

Step 1:

A relation's output can be renamed using the RENAME method. Breaking a difficult chain of operations and renaming it as a relation with new names is sometimes straightforward and appropriate.

Step 2:

A relation's output can be renamed using the RENAME method. It's possible that we'll wish to save the outcome of a relational algebra expression as a relation so we may use it later.